The 1:13 Phased Demultiplexer Circuit

Dr. SERAFIM PORIAZIS
Phasetronic Laboratories
6 Depasta Str., 17122, N.Smirni, Athens
GREECE

Abstract: - The behavior of the 1:13 Phased Demultiplexer (PDMUX13) circuit is analyzed. The circuit demultiplexes the input clock signal into thirteen phased output signals by streaming sets of twenty-six clock phases. A phase difference equal to the half period of the clock is maintained between consecutive output transitions. The VHDL description of the PDMUX13 cell is given and the simulation and synthesis results are generated. A 2-level tree-like structure is built by applying the phased outputs of the PDMUX13 cell into the corresponding clock inputs of thirteen cell replicas that extend the circuit behavior. The EXOR13 gate is attached to the PDMUX13 cell output ports and is aggregating all the phases that the phased clock signals are carrying while preserving their phase associations.

Key-Words: - Circuit design, Clock, Demultiplexer, EXOR aggregation, FPGA, Phased, VHDL

1 Introduction

Demultiplexer (DMUX) ICs chips that target SONET OC-768 applications are able to operate error free beyond 50Gb/s [1]. The given 1:4 DMUX for Serial Communication Systems uses a tree architecture with a recursive series of 1:2 demultiplexer stages. It uses a half-rate clock for the first 1:2 demultiplexing stage.

The 1:4 DMUX with the Multi-phase Clock (MPC) architecture [2] consists of four parallel latch lines and the TFF generating four-phase clock. The IC was fabricated with InP HEMTs and confirmed to operate at up to 50 Gbit/s with 1.42-W power consumption. This circuit demultiplexes a 4f-(bit/s) input to four parallel f-(bit/s) outputs.

A new phase detector, which can perform a 1:2 data demultiplexing function, is given by [3]. The circuit incorporates flip-flops that are triggered by the falling and rising edges of the clock. They have developed a pulse compensation technique by which they detect whether or not a transition exists during the period by using XOR2 and obtain correct up/down pulses from the half-frequency clock.

The correct behavior of synchronous circuits depends upon the distribution of clock signals to different parts of the circuit. As being examined by [4], state machines can be implemented as synchronous circuits whose bistables and register are each clocked by one of a set of periodic signals. In particular the periodic signals would be phased.

The phase associations between the input clock signal and the thirteen phased output signals produced by the 1:13 Phased Demultiplexer (PDMUX13) circuit are examined in this paper. The subject circuit represents a reliable solution to the challenging problem of synchronizing the individual modules of a multiphase model [5], whose operation adopts a 13-phase timing pattern. The present work is based on the principles of operation of the Two-Phase Twisted Ring Counter (2P-TRC) circuit [6] and is targeting data streaming applications. The unit phase duration that is used throughout the text is equal to the half period of the input clock signal. Tree-like structures based on the demultiplexer circuit are built and their behavior is analyzed. The EXOR operator is associated to the aggregation of the phased outputs produced by the PDMUX13. A transposition mechanism assists us to build a primitive and an expanded PDMUX13/EXOR13 configuration, which fully preserve the phase relationships between the clock and the phased signals at the structural level of circuit design.

2 The PDMUX13 Cell

2.1 The basic cell operation

The fundamental cell, which demultiplexes the clock signal *CLK* of frequency *f* into thirteen phased output signals of frequency *f*/13, is called 1-to-13 Phased Demultiplexer (PDMUX13) circuit and is shown in Figure 1. The thirteen overlapping output signals CLK_1=aDOUT1, CLK_2=aDOUT2, CLK_3=aDOUT3, … , CLK_{13}=aDOUT13 have frequency equal to *f*/13 (period of each CLK_i, i=1,2,3,…,13 equals 13·T, where T the period of *CLK*) with signal CLK_1 leading CLK_2 , CLK_2 leading CLK_3 , … , CLK_{12} leading CLK_{13} by a T/2 phase difference. The logic-‘1’ or logic-‘0’ pulse width of each of the above phased signals is equal to 13·T/2. Consecutive changes of logic value at each signal CLK_i, for i=1,2,3,…,13 occur alternatively at the rising and falling edges of *CLK* at a distance of 13·T/2.

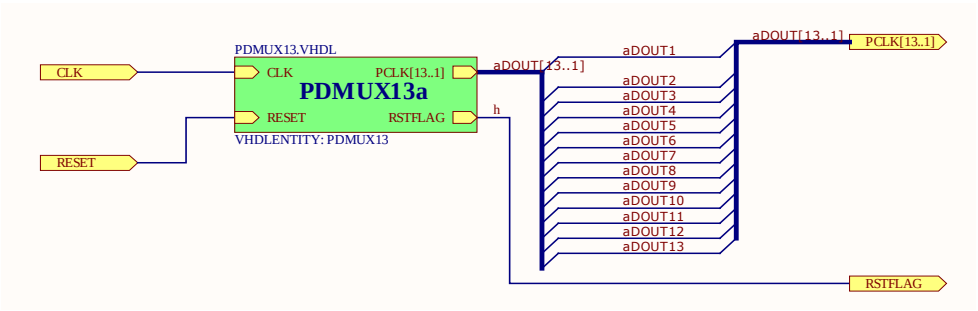

Fig. 1 The PDMUX13 basic cell block diagram

2.2 The valid codeword sequence

When the clock signal *CLK* is applied to the circuit, the following cyclic sequence of codewords is presented at the outputs: CLK_1,CLK_2,CLK_3, … , CLK_{13}= 0000000000000→1000000000000→1100000000000→1110000000000→1111000000000→1111100000000→1111110000000→1111111000000→1111111100000→1111111110000→1111111111000→1111111111100→1111111111110→1111111111111→0111111111111→0011111111111→0001111111111→0000111111111→0000011111111→0000001111111→0000000111111→0000000011111→0000000001111→0000000000111→0000000000011→0000000000001, which is considered as being the normal circuit operation. Each codeword remains stable for the state time of the circuit, that is T/2, and the above sequence is repeated throughout the operation of the cell. Thus the cycle time for the output pattern is defined by the twenty-six-tuple of codewords of length equal to 13·T. This duration forms the period of each phased output signal.

The additional 8166 codewords out of the total 8192 possible codewords that are not included in the above cyclic sequence should be considered during the design of the circuit for achieving reliable operation of the PDMUX13 cell. If the circuit reaches any of these 8166 invalid codewords, then an invalid codeword flag is set at the output of the cell. This flag always forces the reset input ports of cascaded PDMUX13 cells to operate in the reset state, which maintains the proper initializing behavior until a valid codeword appears on the output port of the cell.

2.3 The algorithm aspects of cell operation

The PDMUX13 cell operation is implemented by the VHDL description shown in Figure 2. The VHDL entity section has an input port CLK on which the clock signal of frequency *f* is applied and an input port RESET on which a reset flag is applied. The phased output signals of frequency *f*/13 of the cell are assigned to port PCLK[13..1] of width thirteen, that is PCLK1=aDOUT1, PCLK2=aDOUT2, … , PCLK13= aDOUT13. The output port RSTFLAG is signaling the invalid codeword status of PCLK, or the cell reset state, which suspends the streaming of phases from CLK toward the outputs of the cell. The VHDL architecture section is of type "behavioral" and utilizes a state machine model, where two internal registers are being used, reg1 and reg2, one for the present state named "present_state1" clocked by the rising edge of the clock and the other for the present state named "present_state2" clocked by the falling edge of the clock, respectively. The next state logic block and the ouput logic

```
1   library IEEE;
2   use ieee.std_logic_1164.all;
3
4   entity PDMUX13 is
5     port (CLK , RESET : in std_logic;
6           RSTFLAG : out std_logic;
7           PCLK : out std_logic_vector(13 downto 1));
8   end PDMUX13;
9
10  architecture behavioral of PDMUX13 is
11      type validcodewords is array(1 to 26) of std_logic_vector(13 downto 1);
12      constant phased_output : validcodewords :=
13      ( ( '0', '0', '0', '0', '0', '0' ,'0', '0', '0', '0', '0', '0', '0'), ( '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1'),
14        ( '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '1'), ( '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '1', '1'),
15        ( '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '1', '1'), ( '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '1', '1', '1'),
16        ( '0', '0', '0', '0', '0', '0', '0', '0', '1', '1', '1', '1', '1'), ( '0', '0', '0', '0', '0', '0', '0', '1', '1', '1', '1', '1', '1'),
17        ( '0', '0', '0', '0', '0', '0', '1', '1', '1', '1', '1', '1', '1'), ( '0', '0', '0', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1'),
18        ( '0', '0', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1'), ( '0', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1'),
19        ( '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1'), ( '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1'),
20        ( '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0'), ( '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '0'),
21        ( '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '0'), ( '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '0', '0'),
22        ( '1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '0', '0', '0'), ( '1', '1', '1', '1', '1', '1', '1', '0', '0', '0', '0', '0', '0'),
23        ( '1', '1', '1', '1', '1', '1', '0', '0', '0', '0', '0', '0', '0'), ( '1', '1', '1', '1', '1', '0', '0', '0', '0', '0', '0', '0', '0'),
24        ( '1', '1', '1', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0'), ( '1', '1', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0'),
25        ( '1', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0'), ( '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0') );
26        signal index : integer := 0;
27        signal present_state1, present_state2, next_state: std_logic_vector(13 downto 1);
28        signal invalidcode_flag : std_logic := '0';
29      begin
30        reg1 : process (CLK, RESET)
31          begin
32            if RESET = '1' then present_state1 <= phased_output(1);
33            elsif (CLK='1' and CLK'event ) then present_state1 <= next_state ;
34            end if;
35        end process;
36        reg2 : process (CLK, RESET)
37          begin
38            if RESET = '1' then present_state2 <= phased_output(14);
39            elsif (CLK='0' and CLK'event ) then present_state2 <= next_state ;
40            end if;
41        end process;
42        next_state_logic : process (CLK, RESET)
43          begin
44            case CLK is
45             when '1' => if RESET = '1' then index <= 1; else index <= index + 1; end if;
46             when '0' => if RESET = '1' then index <= 14; else index <= index + 1; end if;
47             when others => null;
48            end case;
49            if index < 26 then next_state <= phased_output(index + 1);
50            else next_state <= phased_output(1); index <= 1; end if;
51            for i in 1 to 26 loop
52             if next_state = phased_output(i) then invalidcode_flag <= '0'; exit;
53             else invalidcode_flag <= '1'; end if;
54            end loop;
55        end process;
56        output_logic : process (index, present_state1, present_state2)
57          begin
58            case CLK is
59             when '1' => PCLK <= present_state1;
60                         if RESET = '1' then RSTFLAG <= '1'; else
61                         RSTFLAG <= invalidcode_flag; end if;
62             when '0' => PCLK <= present_state2;
63                         if RESET = '1' then RSTFLAG <= '1'; else
64                         RSTFLAG <= invalidcode_flag; end if;
65            when others => null;
66            end case;
67        end process;
68  end behavioral;
```

Fig. 2 The VHDL description of the PDMUX13 basic cell

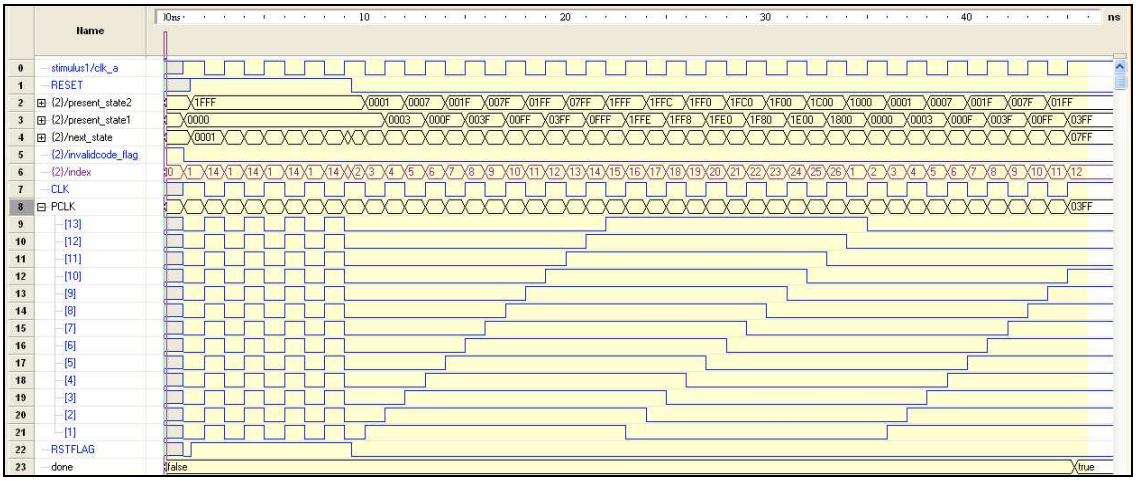

Fig. 3 The PDMUX13 basic cell operation (VHDL simulation results)

block of the model are specified by the corresponding processes "next_state_logic" and "output_logic". The set of twenty-six valid codewords of the circuit are stored in an indexed array of size 26*13=338 bits, that is represented by the constant named "phased_output". The index of the above array cycles through the integer values 1 to 26 specifying the valid codeword entry for the next state signal. Whenever the RESET input is set, that is RESET='1', each output signal from PCLK[13..1] can cycle only through the values "0000000000000" and "1111111111111" thus assuring proper initialization of the circuit at either the rising or the falling edge of CLK.

2.4 The VHDL simulation and synthesis

The VHDL testbench simulation results for the PDMUX13 cell are given in Figure 3. The duration of this simulation is defined by the value of the signal "done". Each of the signals "next_state", "present_state1", "present_state2" and PCLK have each a width of 13 bits. Each value shown on these signal waveforms is hexadecimal. The output port PCLK is analyzed into thirteen individual output signals with waveforms that verify the correct operation of the circuit. The "index" signal has decimal values and defines the index value of the array of valid codewords. The logic value changes of PCLK occur at each rising and at each falling edge of the input signal CLK.

The synthesis of the PDMUX13 cell targeting an FPGA device was successfully performed giving us the following results:

- flip flops with asynchronous reset = 13
- flip flops with asynchronous preset = 13
- combinational feedback paths = 32
- macrocell instances = 1
- combinational logic area estimate = 168 LUTs

3 The Phase Associations of the PDMUX13 Signals

We examine the associations between the input clock signal CLK and the phased output signals PCLK1=CLK_1 , PCLK2= CLK_2 , … , PCLK13=CLK_{13} of the PDMUX13 cell. It is evident that the EXOR function can be utilized to express the following relationships:

$$\begin{aligned} CLK &= CLK_1 \oplus CLK_2 \oplus CLK_3 \oplus ... \oplus CLK_{13} \\ CLK_1 &= CLK \oplus CLK_2 \oplus CLK_3 \oplus ... \oplus CLK_{13} \\ &... \\ CLK_{13} &= CLK \oplus CLK_1 \oplus CLK_2 \oplus ... \oplus CLK_{12} \end{aligned} \qquad (2)$$

The equation (2) satisfies an extended transposition mechanism for the EXOR operator that states the following: "if $f=g\oplus h$, then $g=f\oplus h$ and $h=g\oplus f$ " where f, g and h are binary variables. Thus an EXOR13 gate can be attached to the outputs of the PDMUX13 cell in order to aggregate the phased signals CLK_1 , CLK_2 , CLK_3 , … , CLK_{13} and produce a replica of the input clock signal, that is $CLK\_OUT = CLK$.

A primitive PDMUX13/EXOR13 configuration is thus formed by the two modules, the PDMUX13 cell and the EXOR13 gate, which are being interconnected via the phased signals CLK_1 , CLK_2 , CLK_3 , … , CLK_{13}. All referenced signals of this configuration maintain the transposition mechanism stated above. We notice that the outputs CLK_1, CLK_2, CLK_3, … , CLK_{13} of the PDMUX13 are themselves periodic signals that can be used to drive in succession replicas of the cell, thus forming a tree-like structure. A new phased signal pattern of period $169 \cdot T$ is thus produced with output signals w_1, w_2, w_3 , …, w_{169} (b1DOUT1, b2DOUT1, b3DOUT1,…, b13DOUT13) with w_i leading w_{i+1} (i=1,2,3,…,168) having a phase difference equal to T/2. This structure (PDMUX13*13) is composed of two levels of cells as shown in Figure 4. According to the frequency of the driving signals, the first level is driven by CLK of frequency f and the second level is driven by y_1=aDOUT1, y_2 =aDOUT2, … , y_{13}=aDOUT13 of frequency

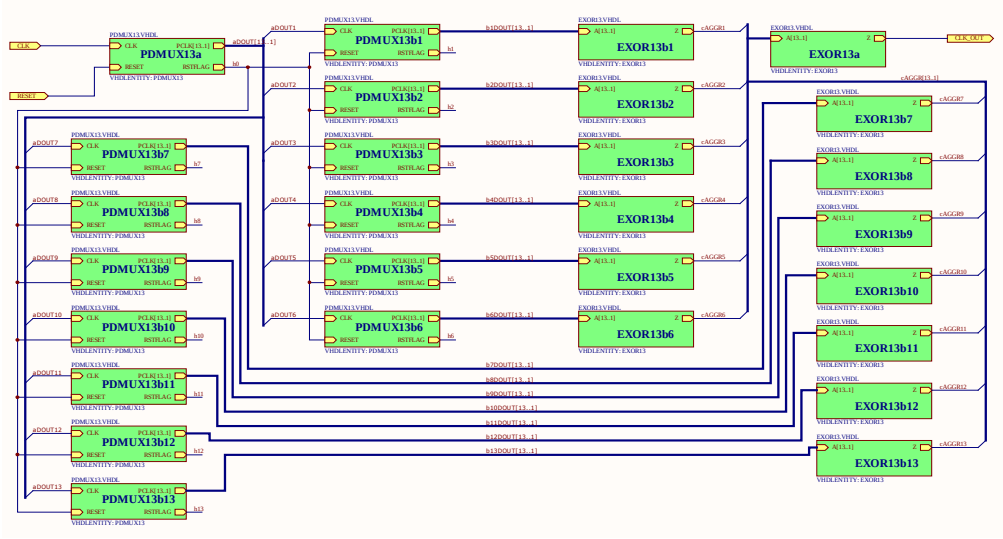

Fig. 4 The 2-level expanded PDMUX13/EXOR13 circuit configuration (block diagram)

f/13. By using an inverse-tree-like structure of EXOR13 gates we can apply the EXOR function to the pattern of the phased signals w_1, w_2, w_3 , … , w_{169} and form a 2-level expanded PDMUX13/EXOR13 configuration. Its aggregated output signal *CLK_OUT* is a replica of the input clock signal *CLK* , thus giving $w_1 \oplus w_2 \oplus w_3 \oplus \ldots \oplus w_{169} = CLK\_OUT = CLK$ which have been verified by the associated simulation results.

The synthesis of the 2-level expanded PDMUX13/EXOR13 circuit configuration targeting an FPGA device was successfully performed giving us the following results:

- flip flops with asynchronous reset = 182
- flip flops with asynchronous preset = 182
- combinational feedback paths = 448
- macrocell instances = 14
- combinational logic area estimate = 2422 LUTs

4 Conclusion

The 1:13 Phased Demultiplexer (PDMUX13) circuit is the basic cell being considered in this paper. Its operation is analyzed and the twenty-six valid codewords that appear at the phased output signals cycle in a predetermined sequence of length 13·T, where T the period of the input clock signal *CLK*. The VHDL description of the PDMUX13 cell is given. The corresponding simulation results verify the proper circuit behavior with the phased output signals PCLK[13..1] maintaining the specified phase associations with *CLK*. The synthesis results are given targeting an FPGA device.

A 2-level expanded PDMUX13/EXOR13 configuration is formed by tree-like structures of the basic cell and the EXOR gate. The input *CLK* is demultiplexed into 169 phased signals which are then aggregated into an output *CLK_OUT* by preserving all the embedded phase associations of the above signals. The simulation results verify the circuit behavior of the 2-level expanded configuration, while the corresponding synthesis results show the complexity requirements for circuit implementation.

References:

[1] M. Meghelli, A.V. Rylyakov, and Lei Shan, *"50 Gb/s SiGe BiCMOS 4:1 multiplexer and 1:4 demultiplexer for serial communication systems,"* in Proceedings of the Solid-State Circuits Conference, 2002, Digest of Technical Papers, ISSCC, 2002 IEEE International, San Francisco, CA, USA, pp 460-465, vol. 1, 3-7 February 2002.

[2] K. Sano, K. Murata, H. Kitabayashi, S. Sugitani, H. Sugahara, and T. Enoki, *"1.4-W 50-Gbit/s InP HEMT 1:4 Demultiplexer IC with a Multi-phase Clock Architecture,"* in Proceedings of the Microwave Symposium Digest, 2003 IEEE MTT-S International, pp 1181-1184, Vol. 2, 8-13 June 2003.

[3] K. Nakamura, M. Fukaishi, H. Abiko, A. Matsumoto, and M. Yotsuyanagi, *"A 6 Gbps CMOS Phase Detecting DEMUX Module Using Half-Frequency Clock,"* in Proceedings of the VLSI Circuits, 1998, Digest of Technical Papers, 1998 Symposium on, Honolulu, HI, USA, pp 196-197, 11-13 June 1998,

[4] S. Poriazis, *Theory and Design of Multiphase Synchronous Circuits*, PhD thesis, Cranfield University, College of Aeronautics, Electronic System Design, England, 1994.

[5] S. Poriazis, *Logic Design of Multiphase Finite State Machines*, Monograph, Phasetronic Laboratories, http://www.phasetroniclab.com Athens, Greece, 2001.

[6] S. Poriazis, *"The Two-Phase Twisted-Ring Counter Circuit,"* in Proceedings of the 2002 IEEE International Symposium on Circuits and Systems (ISCAS'02), Phoenix, Arizona, USA, vol. IV, pp. 858-861, 26-29 May, 2002.